UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,311	09/30/2003	Frank G. Gates	42P16521	8182

8791 7590 01/21/2009

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
RAMPURIA, SATISH	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
01/21/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/676,311	GATES ET AL.	
	Examiner	Art Unit	
	SATISH RAMPURIA	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 November 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-10,16-26,29-34,41,43-47,49,50 and 53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-10,16-26,29-34,41,43-47,49,50 and 53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Response to Amendment

1. This action is in response to the amendment filed on 11/26/2007.

2. The rejections under 35 U.S.C. §112 second paragraph to claim 21-31, 47, and 49 is withdrawn in view of Applicant's amendment.

3. Claims cancelled by the applicant: 2, 11-15, 27, 28, 35-40, 42, 48 and 51-52.

4. Claims amended by the applicant: 21, 29-30, 32, and 46.

5. Claims 1, 3-10, 16-26, 29-34, 41, 43-47, 49, 50, and 53 are pending.

Response to Arguments

6. Applicant's arguments filed 11/26/2007 have been fully considered but they are not persuasive. Since the office action had a typographical error of including Schmidt in the rejection of claims 21, 22, 25, 26, 29-31, 46, 47 and 49. This office action is non-final.

(A) In response to Applicants arguments with respect claim 1 that the combination of Goebel and Schmidt fails to disclose, teach, or suggest executing intermediate code based on external execution input to generate data indicating performance of the intermediate code. Examiner respectfully disagrees. As acknowledge by the office action that Goebel does not explicitly discloses executing intermediate code based on external execution input to generate data indicating performance of the intermediate code. However, the combination of Goebel and Schmidt disclose as indicated in the office action (see the rejection below) that

the limitation as agued by the applicants above is taught by Schmidt. Further, Schmidt explicitly discloses that the intermediate representation code 115 or 516 can be implemented using external execution input to generate data indicating performance of the intermediate code. Front-end compiler 610 could read the source code 105 from the **first computer system**, generate the intermediate representation 615, and store the intermediate representation 615 on **a third computer system**. Back-end compiler 620 could be executed on **a fourth computer system**, which **reads the intermediate representation 615 from the third computer system**, and **generates therefrom machine code 625**, which could be written **to a fifth computer system**, see paragraph [0057].

(B) In response to Applicants arguments with respect amended claims 21 and 46 that the combination of Goebel, Schmidt, and Shupak fails to disclose, teach, or suggest determining whether to perform an additional iteration to produce further modified intermediate and machine code based upon whether the previous iteration achieved a predetermined performance gain. Examiner respectfully disagrees. The office has a typographical error of including Schmidt in the rejection of claims 21, 22, 25, 26, 29-31, 46, 47 and 49. Schmidt was not used in the rejection. As acknowledge by the office action that Goebel does not explicitly discloses the limitations as argued by the applicants. However, the combination of Goebel and Shupak discloses the limitations as argued. More

particularly, Shupak discloses iteratively: determining whether to produce further modified intermediate and further modified machine code based upon whether a predetermined gain has been achieved in the modified code over the machine code (col. 11, lines 1-5 "method 700 includes modifying the executable program in accordance with the annotation debug information (i.e., based upon predetermined gain) and See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710…the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from…the annotation debug information includes information". Since no explicit definition is provided for predetermined gain examiner interprets it broadly i.e., Shupak system modifies code based upon annotation information"); and if the further modified intermediate code and the further modified machine code are to be produced (col. 11, lines 6-10 "modifying the executable program 720, the modifying includes inserting code into the executable program to perform an action in accordance with the information in the annotation debug information"): providing the modified machine code to the profiler (col. 10, lines 48-51 "an output of the computer program analysis tool is ready as input to a second computer program analysis tool and the second computer program analysis tool is a profiler").

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

8. Claims 32-34 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 32 directed to apparatus of functional descriptive material per se, and hence non-statutory. There are no indications or suggestions in the specification [published specification [0023-0025]] or claims that would associate the recited software components in the claims with hardware (e.g., a memory and a processor) elements of the electronic device. The recited components of the claims can reasonably be interpreted as computer program modules/software per se. Therefore, the claims constitute computer programs representing computer listings per se. Such descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable storage device encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer

program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claims 33-34 are directly or indirectly dependent on independent claim 32, thus suffering the similar deficiency as independent claim 32.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1, 3-10, 16-20, 32-34, 41,43-46 and 50,53 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,289,505 to Goebel (hereinafter, Goebel) in view of US Publication No. 2003/0051234 to Schmidt (hereinafter, Schmidt).

Per claim 1:

Goebel discloses:

A method comprising:

- receiving source code (See FIG. 3, element 301 and related discussion);

- transforming the source code to intermediate code (FIG. 3, element 305 and related discussion);

- executing the intermediate code (See FIG. 3, element 303 and 305 and related discussion);

- generating data that indicates performance of the executed intermediate code (See FIG. 3, element 307 and related discussion); and
- producing machine code based on the data and intermediate code (See FIG. 3, elements 305-309 and FIG. 4, element 411 and related discussion).

Goebel does not explicitly disclose executing an intermediate code based on external execution unit.

However, Schmidt discloses in an analogous computer system executing an intermediate code based on external execution unit (paragraph [0040] "Back-end compiler 620 processes the intermediate representation 615, and generates therefrom machine code 625. Note that the architected division 660 between front-end compiler 610 and back-end compiler 620 is maintained, allowing any suitable front-end compiler 610 to generate intermediate representation code 615 for any compatible back-end compiler 620").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of executing an intermediate code based on external execution unit as taught by Schmidt into the method of re-optimizing a previously compiled binary executables as taught by Goebel. The modification would be

obvious because of one of ordinary skill in the art would be motivated to executing an intermediate code based on external execution unit to provide an optimized method to generate executable code for computer systems as suggested by Schmidt (paragraph [0008]).

Per claim 3:

The rejection of claim 2 is incorporated and further, Goebel discloses:

- wherein executing the intermediate code comprises simulating execution of the intermediate code (See FIG. 4, element 409 and related discussion).

Per claim 4:

The rejection of claim 2 is incorporated and further, Goebel discloses:

- wherein generating the data regarding the performance of the executed intermediate code comprises generating a performance profile (See FIG. 4, element 407 and related discussion).

Per claim 5:

The rejection of claim 4 is incorporated and further, Goebel discloses:

- wherein generating the data regarding the performance of the executed intermediate code further comprises annotating the intermediate code based, at least in part, on

performance profile data (See FIG. 3, elements 305-309 and FIG. 4, element 411 and related discussion and col. 6, lines 45-67 "process profile information… generated during execution…" and col. 7, lines 3-52 "…processes…annotation information… was generated during the compilation… ").

Per claim 6:

The rejection of claim 5 is incorporated and further, Goebel discloses:

- wherein annotating the intermediate code comprises concatenating data structures that include the performance profile data to intermediate code to embed the performance profile data into the intermediate code (See FIG. 3, elements 315, 317 and related discussion).

Per claim 7:

The rejection of claim 5 is incorporated and further, Goebel discloses:

- wherein annotating the intermediate code comprises: generating a file that includes the performance profile data; and mapping the performance profile data to corresponding portions of intermediate code (col. 6, lines 45-67 "process profile information… generated during execution…" and col. 7, lines 3-52

"…processes…annotation information… was generated during the compilation… ").

Per claim 8:

The rejection of claim 5 is incorporated and further, Goebel discloses:

- wherein producing machine code based on the data and intermediate code (See FIG. 3, elements 305-309 and FIG. 4, element 411 and related discussion) includes providing the annotated intermediate code to a compiler, wherein the compiler produces the machine code based on annotated intermediate code (See FIG. 3, elements 305-309 and FIG. 4, element 411 and related discussion and col. 6, lines 45-67 "process profile information… generated during execution…" and col. 7, lines 3-52 "…processes…annotation information… was generated during the compilation… ").

Per claim 9:

The rejection of claim 5 is incorporated and further, Goebel discloses:

- wherein the performance profile data comprises one or more of branch statistics, loop statistics and function invocation statistics (col. 7, lines 10-16"…optimization include…techniques for interprocedural optimization and local, loop, and global scheduling…").

Per claim 10:

The rejection of claim 8 is incorporated and further, Goebel discloses:

- wherein the machine code executes faster than the intermediate code (See FIG. 3, elements 315, 317 and related discussion).

Per claim 16:

The rejection of claim 1 is incorporated and further, Goebel discloses:

- receiving external execution input (See FIG. 3, element 313 and related discussion); and
- using the external execution input to execute the intermediate code (See FIG. 3, element 305 and related discussion).

Per claim 17:

The rejection of claim 1 is incorporated and further, Goebel discloses:

- wherein the data comprises one or more of plain-text format, binary representations, database maps, and character delimited proprietary format (col. 8 to col. 9, lines65-67 and 1-3 "…save data set specific binary executable…optimized binary executable on the data set to achieve the optimized performance with respect to that data set…").

Per claims 18 and 20:

Goebel discloses:

- A method comprising: transforming source code into intermediate code (See FIG. 3, element 315 and related discussion);
- providing the intermediate code to a profiler that executes the intermediate code and generates annotated intermediate code based on the performance of the executed intermediate code (col. 6, lines 45-67 "process profile information… generated during execution…" and col. 7, lines 3-52 "…processes…annotation information… was generated during the compilation… " and See FIG. 4, element 407 and related discussion);
- receiving from the profiler the annotated intermediate code (See FIG. 3, element 305 and related discussion); and
- transforming the annotated intermediate code into machine code (See FIG. 4, element 411 and related discussion).

Goebel does not explicitly disclose providing the intermediate code to a profiler that *executes the intermediate code based on external execution input.*

However, Schmidt discloses in an analogous computer system providing the intermediate code to a profiler that *executes the intermediate code based on external execution input* (paragraph [0040] "Back-end compiler 620 processes the intermediate representation 615, and generates therefrom machine code 625. Note that the architected division 660 between front-end compiler 610 and back-end compiler 620 is maintained, allowing any suitable

front-end compiler 610 to generate intermediate representation code 615 for any compatible back-end compiler 620").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of providing the intermediate code to a profiler that *executes the intermediate code based on external execution input* as taught by Schmidt into the method of re-optimizing a previously compiled binary executables as taught by Goebel. The modification would be obvious because of one of ordinary skill in the art would be motivated to providing the intermediate code to a profiler that *executes the intermediate code based on external execution input* to provide an optimized method to generate executable code for computer systems as suggested by Schmidt (paragraph [0008]).

Per claim 19:

The rejection of claim 18 is incorporated and further, Goebel discloses:

- wherein the annotated intermediate code is annotated to include one or more of branch statistics, loop statistics and function invocation statistics (col. 7, lines 3-52 "…optimization include…techniques for interprocedural optimization and local, loop, and global scheduling…").

Claims 32-34 are the apparatus claim corresponding to method claims 1, 19-20 respectively, and rejected under the same rational set forth in connection with the rejection of claims 1, 19-20 respectively, above.

Claims 41, 43-45 are the article of manufacture claim corresponding to method claims 1, 2, 5, 14, and 8 respectively, and rejected under the same rational set forth in connection with the rejection of claims 1, 2, 5, 14, and 8 respectively, above.

Claims 50, 53 are the system claim corresponding to method claims 1, and 16 respectively, and rejected under the same rational set forth in connection with the rejection of claims 1 and 16 respectively, above.

11. Claims 21, 22, 25, 26, 29-31, 46, 47 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Goebel in view of US Patent No. 6,874,410 to Shupak (hereinafter, Shupak).

Per claim 21:

Goebel discloses:

- A method comprising:
- Producing intermediate code and machine code based upon source code (See FIG. 1 and related discussion);
- receiving a data file generated by a profiler (See FIG. 3 element 317 and related discussion), wherein the data file indicates a performance of the machine code as executed by the profiler (See FIG. 3, element 307 and related discussion); and
- producing modified intermediate code and machine code based on the source code and the data file; (See FIG. 3, elements 305-309 and related discussion) and ;
- receiving another data file from the profiler; and producing further modified intermediate and machine code based upon the source code and the another data file (col. 6, lines 45-67 "process profile information… generated during execution…" and col. 7, lines 10-16 "…processes…annotation information… was generated during the compilation…").

Goebel does explicitly disclose iteratively: determining whether to produce further modified intermediate and further modified machine code based upon whether a predetermined gain has been achieved in the modified code over the machine code; and if the further modified intermediate code and the further modified machine code are to be produced: providing the modified machine code to the profiler.

However, Shupak discloses in an analogous computer system iteratively: determining whether to produce further modified intermediate and further modified machine code based upon whether a predetermined gain has been achieved in the modified code over the machine code (col. 11, lines 1-5 "method 700 includes modifying the executable program in accordance with the annotation debug information (i.e., based upon predetermined gain) and See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710…the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from…the annotation debug information includes information". Since no explicit definition is provided for predetermined gain examiner interprets it broadly i.e., Shupak system modifies code based upon annotation information"); and if the further modified intermediate code and the further modified machine code are to be produced (col. 11, lines 6-10 "modifying the executable program 720, the modifying includes inserting code into the executable program to perform an action in accordance with the information in the annotation debug information"): providing the modified machine code to the profiler (col. 10, lines 48-51 "an output of the computer program analysis tool is ready as input to a second computer program analysis tool and the second computer program analysis tool is a profiler").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of iteratively determining whether to produce further modified intermediate and further modified machine code based upon whether a predetermined gain has been achieved in the modified code over the machine code; and if the further modified intermediate code and the further modified machine code are to be produced providing the modified machine code to the profiler as taught by Shupak into the method of generating the optimized executable code as taught by Goebel. The modification would be obvious because of one of ordinary skill in the art would be motivated iteratively determining whether to produce further modified intermediate and further modified machine code based upon whether a predetermined gain has been achieved in the modified code over the machine code; and if the further modified intermediate code and the further modified machine code are to be produced and providing the modified machine code to the profiler to avoid the overhead execution time as suggested by Shupak (col. 3, lines 25-40).

Per claim 22:

The rejection of claim 21 is incorporated and further, Goebel discloses:

- providing the machine code to the profiler (See FIG. 4 and related discussion).

Per claim 25:

The rejection of claim 21 is incorporated and further, Goebel discloses:

- wherein the data file includes one or more of branch statistics, loop statistics and function invocation statistics (col. 7, lines 10-16 "…optimization include…techniques for interprocedural optimization and local, loop, and global scheduling…").

Per claim 26:

The rejection of claim 21 is incorporated and further, Goebel discloses:

- wherein the data file includes an identifier that associates an executed instruction with generated data (col. 8 to col. 9, lines 65-67 and 1-3 "…save data set specific binary executable…optimized binary executable on the data set to achieve the optimized performance with respect to that data set…").

Per claim 28:

The rejection of claim 21 is incorporated and further, Goebel does not explicitly disclose wherein determining whether to produces further modified machine code comprises determining whether a predetermined performance gain has been achieved.

However, Shupak discloses in an analogous computer system wherein determining whether to produces further modified machine code comprises determining whether a predetermined performance gain has been achieved (See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710...the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from...the annotation debug information includes information").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of wherein determining whether to produces further modified machine code comprises determining whether a predetermined performance gain has been achieved as taught by Shupak into the method of generating the optimized executable code as taught by Goebel. The modification would be obvious because of one of ordinary skill in the art would be motivated to produce the modified code if it is need to be modified to avoid the overhead execution time as suggested by Shupak (col. 3, lines 25-40).

The feature of wherein determining whether to produces further modified machine code comprises determining whether a predetermined performance gain has been achieved would be obvious for the reasons set forth in the rejection of claim 21.

Per claim 29:

The rejection of claim 28 is incorporated and further, Goebel does not explicitly disclose wherein determining whether the predetermined performance gain has been achieved comprises determining whether the modified machine code executes faster than the machine code.

However, Shupak discloses in an analogous computer system wherein determining whether the predetermined performance gain has been achieved comprises determining whether the modified machine code executes faster than the machine code (col. 11, lines 1-5 "method 700 includes modifying the executable program in accordance with the annotation debug information (i.e., based upon predetermined gain) and See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710...the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from...the annotation debug information includes information").

The feature of wherein determining whether the predetermined performance gain has been achieved comprises determining whether the modified machine code executes faster than the machine code would be obvious for the reasons set forth in the rejection of claim 21.

Per claim 30:

The rejection of claim 28 is incorporated and further, Goebel does not explicitly disclose wherein determining whether to produces further modified machine code comprises determining whether a cost of modifying the modified machine code exceeds a performance gain to be achieved by the modifying.

However, Shupak discloses in an analogous computer system wherein determining whether to produces further modified machine code comprises determining whether a cost of modifying the modified machine code exceeds a performance gain to be achieved by the modifying (See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710...the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from...the annotation debug information includes information").

The feature of wherein determining whether to produces further modified machine code comprises determining whether a cost of modifying the modified machine code exceeds a performance gain to be achieved by the modifying would be obvious for the reasons set forth in the rejection of claim 21.

Per claim 31:

The rejection of claim 21 is incorporated and further, Goebel does not explicitly disclose wherein receiving the data file comprises receiving the data file via one of a data

storage device, an alphanumeric input device, a network interface, a shared data storage location, and a direct real-time connection.

However, Shupak discloses in an analogous computer system wherein receiving the data file comprises receiving the data file via one of a data storage device, an alphanumeric input device, a network interface, a shared data storage location, and a direct real-time connection (See FIG. 7 and related discussion col. 10, lines 61-67 "Method 700 includes receiving or reading an annotation debug information in an executable computer program 710...the annotation debug information was generated from an annotation function call in the source code that the executable computer program was compiled from...the annotation debug information includes information").

The feature of wherein receiving the data file comprises receiving the data file via one of a data storage device, an alphanumeric input device, a network interface, a shared data storage location, and a direct real-time connection would be obvious for the reasons set forth in the rejection of claim 21.

Claims 46, 47, 49 are the article of manufacture claim corresponding to method claims 21, 29, 31 respectively, and rejected under the same rational set forth in connection with the rejection of claims 21, 29 and 31 respectively, above.

12. Claims 23 and 24 rejected under 35 U.S.C. 103(a) as being unpatentable over Goebel in view of Shupak and further in view of Applicant's Admitted Prior Art (hereinafter, AAPA).

Per claim 23:

The rejection of claim 22 is incorporated and further, neither Goebel nor Shupak explicitly disclose wherein providing the machine code to the profiler comprises providing the machine code to a virtual machine.

However, AAPA discloses in an analogous computer system wherein providing the machine code to the profiler comprises providing the machine code to a virtual machine (Applicant's Specification [0027] "Virtual machine are known in the art, and thus will not be described further…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of wherein providing the machine code to the profiler comprises providing the machine code to a virtual machine as taught by AAPA into the method of generating the optimized executable code as taught by the combination system of Goebel and Shupak. The modification would be obvious because to of one of ordinary skill in the art providing the machine code to the profiler

comprises providing the machine code to a virtual machine would be known as suggested by AAPA (page 11 [0027]).

Per claim 24:

The rejection of claim 22 is incorporated and further, neither Goebel nor Shupak explicitly disclose wherein providing the machine code to the profiler comprises providing the machine code to a probed processor.

However, AAPA discloses in an analogous computer system wherein providing the machine code to the profiler comprises providing the machine code to a probed processor (Applicant's Specification [0028] "Probed hardware is known in the art, and thus will not be described further…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of wherein providing the machine code to the profiler comprises providing the machine code to a probed processor as taught by AAPA into the method of generating the optimized executable code as taught by the combination system of Goebel and Shupak. The modification would be obvious because to of one of ordinary skill in the art providing the machine code to the profiler comprises providing the machine code to a probed processor would be known as suggested by AAPA (page 12 [0028]).

Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday. Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: 571-272-2100**.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer

Art Unit 2191

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191